Theoretical Computer Science 250 (2001) 247–264

Theoretical Computer Science

www.elsevier.com/locate/tcs

Changwook Kim

School of Computer Science, University of Oklahoma, 200 Felgar Street, Norman, Oklahoma 73019, USA

Received September 1997; revised January 1999
Communicated by A. Salomaa

Abstract

Every context-free grammar can be transformed into one in double Greibach operator form, that satisfies both double Greibach form and operator form. Examination of the expressive power of various well-known subclasses of context-free grammars in double Greibach and/or operator form yields an extended hierarchy of language classes. Basic decision properties such as equivalence can be stated in stronger forms via new classes of languages in this hierarchy.

Keywords: Formal languages; Context-free grammars; Normal forms; Hierarchy; Decision properties

1. Introduction

Two well-known normal forms for context-free grammars are the double Greibach normal form [17] in which the right-hand side of each production starts and ends with a terminal symbol (which extends the Greibach normal form [8] that imposes the same condition only on the first symbol of the right-hand side) and the operator normal form [7] in which the right-hand side of each production contains no adjacent nonterminal symbols (see also [9] for these normal forms). The present paper considers various subclasses of context-free grammars in double Greibach and/or operator form and analyzes their expressive power and decision properties. This yields a new normal form for context-free grammars (and derivation-bounded grammars), the double Greibach operator normal form that satisfies both double Greibach form and operator form, and an extended hierarchy of language classes, augmented by subclasses of context-free grammars for which the double Greibach operator form is not a normal form. Decision

E-mail address: kim@cs.ou.edu (C. Kim).

0304-3975/01/$ - see front matter
PII: S0304-3975(99)00138-3

properties known for existing language classes can be restated in stronger forms via new classes in this hierarchy.

Consideration of such restricted classes of grammars has been motivated by the recent intensive study of apex/boundary graph grammars whose rewriting mechanism is similar to the one as contained in the double Greibach/operator form. For example, in the so-called apex node-label-controlled graph grammars [5], only terminal nodes in the right-hand side of a production can be adjacent to outside nodes when rewriting takes place (the double Greibach form property) and this results in the operator form property also (i.e., nonterminal nodes are not adjacent in the right-hand side of each production) since adjacency between nonterminal nodes is irrelevant and can be simply removed. (The operator form property is called boundary or separated property in graph grammar theory. For string grammars, the double Greibach form property does not immediately imply the operator form property since adjacency between symbols cannot be broken arbitrarily, yet we shall see that this property holds.) Various kinds of apex/boundary graph grammars have been studied in the literature, see, e.g., [1, 3–6, 12–16, 18]. Double Greibach (operator) grammars are, in a sense, a special type of apex (boundary) graph grammars that generate chains only. Thus, the present work may be viewed as an attempt to link the currently active research in graph grammar theory and the classical string grammar theory. Such a relationship between two systems may be used, e.g., to transfer known results for subclasses of context-free grammars to graph grammars, and possibly vice versa.

Despite the abovementioned motivation from graph grammar theory, the present paper will focus on string grammars and their languages. Thus, no knowledge in graph grammars is needed to read this paper. After settling some preliminary notations at the end of this section, we shall prove in Section 2 that the double Greibach operator form is a normal form for context-free grammars. This will be done by extending the Engelfriet's transformation for double Greibach normal form, presented recently in [2]. Section 3 examines expressive power of five well-known subclasses of context-free grammars (the right-linear, linear, k-linear, nonterminal-bounded and derivation-bounded grammars) in double Greibach and/or operator form and constructs an extended hierarchy of language classes. Section 4 considers several standard decision properties and presents their decidability/undecidability via new classes in this hierarchy.

For a word x, its length will be denoted by $\#(x)$ and, for a symbol a, the number of a's in x by $\#_a(x)$. The empty word is denoted by λ. The inclusion (proper inclusion) relation between two sets is denoted by $\subseteq$ ($\subset$). A context-free grammar is defined by a 4-tuple $G=(N,\Sigma,P,S)$, where N is the set of nonterminal symbols, Σ is the set of terminal symbols, P is the set of production rules, and $S \in N$ is the start symbol. We can assume without loss of generality that if G generates λ then $S \to \lambda$ is the only λ-production (whose right-hand side is λ) and S does not appear in the right-hand side of any production in G. Thus, as $S \to \lambda$ can be left out during any grammar transformation process and added back later, we shall simply assume that $\lambda \notin L(G)$ when considering grammar transformations. We shall use $\Rightarrow_G$ (or simply $\Rightarrow$) to denote a direct derivation in G and $\Rightarrow_G^*$ (or $\Rightarrow^*$) for its transitive reflexive closure. Other notations not defined in this paper can be found in standard textbooks in formal language theory such as [10, 19].

2. Double Greibach operator normal form

We shall prove that every context-free grammar can be transformed into one in double Greibach operator form, that satisfies both double Greibach form and operator form. This will be done by extending the Engelfriet's transformation for double Greibach normal form, which we shall discuss below in details. Several other results presented in the next section will also need analysis of this method.

The double Greibach normal form for context-free grammars was first proved by Rosenkrantz [17] by using formal power series and matrix techniques. Recently, Engelfriet [2] presented an alternative method for this normal form, which is basically a top-down nondeterministic process of guessing the leftmost and rightmost symbols of the word generated by each nonterminal node in a parse tree and verifying it at the leaves, by extending a similar method used earlier for Greibach normal form and operator normal form (see [9, 20]). This method has also been used recently to transform certain context-free-like graph grammars into the Greibach normal form [3, 6].

The Engelfriet's transformation works as follows. Let $G=(N,\Sigma,P,S)$ be a context-free grammar in Chomsky normal form, i.e., the right-hand side of each production has the form $A\to BC$ or $A\to a$, where $A,B,C\in N$ and $a\in\Sigma$. Let $N'=N\cup\{[A,B],(A,B)\mid A,B\in N\}$ and construct $G'=(N',\Sigma,P',S)$, where P' consists of the following productions:
(1) $A\to a$ if $A\to a$ is in P,
(2) $A\to a[Y,B_1](B_2,Z)b$ if $A\to B_1B_2$, $Y\to a$ and $Z\to b$ are in P,
(3) $[D,A]\to Q[C,B]R$ if $A\to BR$ and $C\to DQ$ are in P,
(4) $[D,A]\to R$ if $A\to DR$ is in P,
(5) $(A,D)\to R(B,C)Q$ if $A\to RB$ and $C\to QD$ are in P,
(6) $(A,D)\to R$ if $A\to RD$ is in P,
(7) $[A,A]\to\lambda$ if $A\in N$,
(8) $(A,A)\to\lambda$ if $A\in N$.

The fact that $L(G')=L(G)$ can be proved by showing that the following three relations hold, by an induction on the length of derivations (the second and third relations are needed to prove the first relation):
(i) $A\Rightarrow_{G'}^* w$ iff $A\Rightarrow_G^* w$,
(ii) $[D,A]\Rightarrow_{G'}^* w$ iff $A\Rightarrow_G^* Dw$,
(iii) $(A,D)\Rightarrow_{G'}^* w$ iff $A\Rightarrow_G^* wD$.
Informally, note that if $A\Rightarrow_G^* w$ then either it is a one-step derivation or there are three productions $A\to B_1B_2$, $Y\to a$ and $Z\to b$ such that $B_1\Rightarrow_G^* Yw_1\Rightarrow_G aw_1$, $B_2\Rightarrow_G^* w_2Z\Rightarrow_G w_2b$ and $w=aw_1w_2b$. This explains the "if" direction of (i) via the productions (1) and (2). The "if" direction of (ii) can be observed by using the productions (3), (4) and (7). Specifically, if $A\Rightarrow_G^* Dw$ then either it is a zero-step derivation or D is a

descendant of A in a derivation tree for $A \Rightarrow_G^* Dw$, and so, either there is a production $A \to DR$ of G such that $A \Rightarrow_G DR \Rightarrow_G^* Dw$ or there are productions $A \to BR$ and $C \to DQ$ of G such that $A \Rightarrow_G BR \Rightarrow_G^* Bw_1 \Rightarrow_G^* Cw_2w_1 \Rightarrow_G DQw_2w_1 \Rightarrow_G^* Dw_3w_2w_1$ $(=Dw)$. The "if" direction of (iii) can be observed similarly by using the productions (5), (6) and (8). Now, the "only if" directions can be observed by using similar arguments.

Note that productions (1) and (2) of G' are already in double Greibach form. Therefore, the usual production substitutions, of replacing each nonterminal symbol from N in the right-hand sides of productions (3)–(6) by its right-hand sides in (1) and (2), in all possible ways, convert the productions (3)–(6) into the double Greibach form. Now, all λ-productions in G' can be removed by using a standard method. In the resulting grammar G'', all productions are in double Greibach form.

Theorem 2.1. *Every context-free grammar can be transformed into an equivalent one in double Greibach operator normal form.*

Proof. Let $G=(N,\Sigma,P,S)$ be a context-free grammar in Chomsky normal form. The Engelfriet's transformation discussed above converts G into G'' in double Greibach normal form. We shall transform G'' further into G''' in double Greibach operator form by repeating a process similar to the Engelfriet's transformation. In fact, the transformation used below is much simpler than the Engelfriet's transformation and is essentially identical to the operator normal form transformation presented in [9].

It is not difficult to observe that G'' contains productions of the following forms only, where upper-case letters are nonterminals and lower-case letters are nonempty terminal words:
(a) $A \to x$,
(b) $A \to xBy$,
(c) $A \to xByCz$,
(d) $A \to xByCzDw$,
(e) $A \to xBCy$,
(f) $A \to xByCDz$,
(g) $A \to xBCyDz$,
(h) $A \to xByCzDEw$,
(i) $A \to xBCyDzEw$,
(j) $A \to xBCyDEz$,
(k) $A \to xBCyDzEFw$.

Productions (a)–(d) are already in the double Greibach operator form. Productions (e)–(k) can be replaced by all productions of the following forms, where a and b are arbitrary terminal symbols:
(e′) $A \to x(B,a)aCy$,
(f′) $A \to xBy(C,a)aDz$,
(g′) $A \to x(B,a)aCyDz$,
(h′) $A \to xByCz(D,a)aEw$,
(i′) $A \to x(B,a)aCyDzEw$,

(j′) $A \to x(B,a)aCy(D,b)bEz$,
(k′) $A \to x(B,a)aCyDz(E,b)bFw$.

Then, productions (e′)–(k′) are also in the desired form. To take care of the nonterminals of the form (A,a) in (e′)–(k′), we add the following productions for each production of the form $A \to \alpha$ in (a)–(d) and (e′)–(k′), where b is an arbitrary terminal symbol:
(l′) $(A,a) \to \gamma(B,b)b$ if $\alpha = \gamma Ba$,
(m′) $(A,a) \to \gamma c$ if $\alpha = \gamma ca$,
(n′) $(A,a) \to \lambda$ if $\alpha = a$.

This covers all productions in G''. Now, all λ-productions can be removed in the usual way. Let G''' be the final grammar obtained by performing the above transformation. It is straightforward to see that G''' is in double Greibach operator form and $L(G''') = L(G'')$ $(= L(G))$. □

3. An extended hierarchy of language classes

In this section, we shall analyze the expressive power of several well-known subclasses of context-free grammars with the additional constraint of double Greibach and/or operator form. In other words, we shall examine whether the expressive power of these known subclasses of context-free grammars is reduced when the double Greibach and/or operator form is additionally imposed and, if so, how much. The results proved in this section will be summarized in the form of an extended hierarchy of context-free language classes.

We shall first define the restrictions of context-free grammars considered in this paper. These restrictions can be found in, e.g., [9, 19].

Definition 3.1. Let $G = (N, \Sigma, P, S)$ be a context-free grammar. Let $A, B \in N$, $x, y \in \Sigma^*$, and $\alpha \in (N \cup \Sigma)^*$. Then, G is
(1) *right-linear* if each production is of the form $A \to xB$ or $A \to x$,
(2) *linear* if each production is of the form $A \to xBy$ or $A \to x$,
(3) *k-linear* (for each fixed $k \geqslant 1$) if each production is of the form $A \to xBy$, $A \to x$ or $S \to \alpha$, where α contains at most k nonterminals and S does not appear in the right-hand side of any production (or *metalinear* if G is k-linear for some k),
(4) *nonterminal-bounded* (or *ultralinear*) if there exists an integer $k \geqslant 1$ such that each sentential form of G contains at most k nonterminals, and
(5) *derivation-bounded* (or *of finite index*) if there exists an integer $k \geqslant 1$ such that each word in $L(G)$ has a derivation in which each sentential form contains at most k nonterminals.

Note that 1-linear grammars are equivalent to linear grammars. The class of languages generated by linear (metalinear) grammars is equivalent to the class of languages accepted by 1-turn (finite-turn) pushdown automata [10]. (For each $k \geqslant 1$, a k-turn pushdown automaton is a pushdown automaton which never makes more than k reversals of its head motion on the pushdown store. A finite-turn pushdown automaton is a k-turn pushdown automaton for some $k \geqslant 1$.) It is known that nonterminal-bounded grammars also characterize the finite-turn pushdown automaton languages [9]. The class of languages generated by derivation-bounded grammars is equivalent to the class of languages generated by the so-called nonexpansive grammars in which no nonterminal A has a derivation of the form $A \Rightarrow^* \alpha$ such that α contains two occurrences of A [19].

Let RL (Lin, k-Lin, NB, and DB) denote the class of languages generated by right-linear (linear, k-linear, nonterminal-bounded, and derivation-bounded) grammars and let CF denote the class of context-free languages. The following relations are known (see [19]).

Theorem 3.2. RL $\subset$ Lin $\subset$ 2-Lin $\subset$ 3-Lin $\subset \cdots \subset$ NB $\subset$ DB $\subset$ CF.

The grammars of various types in double Greibach and/or operator form will be denoted by suffixing DG and/or O to their class names. We shall extend the hierarchy stated in Theorem 3.2 to the one shown in Fig. 1, where arrows denote proper inclusion relations and any two classes not related by a chain of arrows are incomparable. Note that RL-DG (=RL-DGO) contains finite languages only and is not shown in this hierarchy.

Theorem 3.3. RL = RL-O $\subset$ Lin-DGO.

Proof. The relation RL = RL-O is trivially true. To see the other relation in the theorem, let $G = (N, \Sigma, P, S)$ be a right-linear grammar in $(A \to aB, A \to a)$-normal form, where $A, B \in N$ and $a \in \Sigma$. We apply a nondeterministic process similar to the Engelfriet's transformation directly to G. Let G' be the context-free grammar containing the following productions:

$A \to a$	if $A \to a$ is in P,
$A \to a(B, C)b$	if $A \to aB$ and $C \to b$ are in P,
$(A, D) \to a(B, C)b$	if $A \to aB$ and $C \to bD$ are in P,
$(A, B) \to a$	if $A \to aB$ is in P,
$(A, A) \to \lambda$	if $A \in N$.

It is easy to see that $L(G') = L(G)$. If we remove all λ-productions, then G' contains productions of the form $A \to a$, $A \to ab$ and $A \to aBb$ only, where A, B are nonterminals and a, b are terminals. Clearly, G' is a Lin-DGO grammar. Therefore, RL $\subseteq$ Lin-DGO. Now, the proper inclusion relation in the theorem follows from the fact that $\{a^n b^n \mid n \geqslant 1\} \in$ Lin-DGO $-$ RL. □

Theorem 3.4. k-Lin-DGO $- (k-1)$-Lin $\neq \emptyset$ *for all* $k \geqslant 2$.

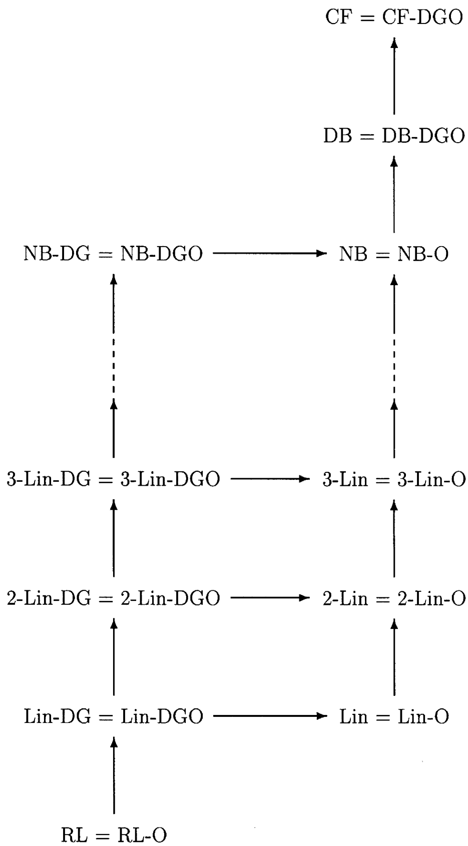

Fig. 1. An extended hierarchy of language classes.

Proof. The language $L_k = \{a^n ba^n c \mid n \geqslant 1\}^k$ is not a $(k-1)$-linear language for all $k \geqslant 2$ [19, Theorem 8.2]. However, L_k is generated by a k-linear grammar $G = (N, \Sigma, P, A)$ such that $N = \{A, B_1, B_2, \ldots, B_k\}$, $\Sigma = \{a, b, c\}$, and P consists of the following productions:

$$A \to aB_1cB_2c \cdots cB_kc,$$

$$B_1 \to aB_1a \mid ba,$$

$$B_i \to aB_ia \mid aba \quad (2 \leqslant i \leqslant k),$$

which is certainly in double Greibach operator form. □

Theorem 3.5. Lin $-$ NB-DGO $\neq \emptyset$.

Proof. Let $L = \{a^n b^{2n} \mid n \geqslant 1\}$ and let s be a regular substitution such that $s(a) = a$ and $s(b) = bc^*$. Then $s(L)$ is a linear language. (It is easy to see that $s(L)$ is accepted by a 1-turn pushdown automaton.) Suppose, to the contrary, that there is a nonterminal-bounded grammar $G=(N,\Sigma,P,S)$ in double Greibach operator form such that $L(G)=s(L)$. Observe first, by a standard pumping argument, that every word $w\in L(G)$ containing sufficiently many a's is generated by using a derivation of the form

$$S\Rightarrow^* uAy\Rightarrow^* uvAxy\Rightarrow^* uvzxy\ (=w),$$

where vx contains at least one a. Let $m=\#_a(vx)$. Then, in fact, $v=a^m$ and $\#_b(x)=2m$. Furthermore, the subderivation $A\Rightarrow^* vAx$ must use linear productions only since, otherwise, G would not be nonterminal-bounded (i.e., if $A\Rightarrow^* \alpha X\beta\Rightarrow^* \alpha v_1Ax_1\beta\Rightarrow^* v_2v_1Ax_1x_2$ $(=vAx)$ and $\alpha\beta$ contains a nonterminal, then $A\Rightarrow^* (\alpha v_1)^iA(x_1\beta)^i$ for all $i\geqslant 0$). Now, let $k=\max\{\#(\beta)\mid B\to\alpha C\beta$ is in P, where $B,C\in N$ and $\alpha,\beta\in\Sigma^*\}$. Then, $k\geqslant 1$. Consider the case where $w=a^n(bc^k)^{2n}$, for a sufficiently large n. Then, the subderivation $A\Rightarrow^* vAx$ takes at most m steps because $v=a^m$ and G is in Greibach form. However, $\#(x)\geqslant 2m+(2m-1)k$ since $\#_b(x)=2m$, and so, $A\Rightarrow^* vAx$ should take at least $\lceil\#(x)/k\rceil\geqslant 2m>m$ steps (because $m\geqslant 1$), a contradiction. It follows that $s(L)\notin$ NB-DGO. As $s(L)\in$ Lin, the theorem holds. $\square$

Theorem 3.6. *The following relations are true*: (1) k-Lin-DG $=$ k-Lin-DGO $\subset$ k-Lin $=$ k-Lin-O *for all* $k\geqslant 1$, *and* (2) NB-DG $=$ NB-DGO $\subset$ NB $=$ NB-O.

Proof. The relation 1-Lin-DG $=$ 1-Lin-DGO is trivially true. To prove k-Lin-DG $=$ k-Lin-DGO for $k\geqslant 2$, let $G=(N,\Sigma,P,S)$ be an arbitrary k-linear grammar in double Greibach form. Then, each production in P has the form $A\to xBy$, $A\to x$ or $S\to x\alpha y$, where $A,B\in N$, $x,y\in\Sigma^+$, α contains at most k nonterminals, and S does not appear in the right-hand side of any production in P. To transform G into the double Greibach operator form, it is sufficient to transform $S\to x\alpha y$ into the right form. This can be done easily by using production substitutions.

A context-free grammar $G=(N,\Sigma,P,S)$ is nonterminal-bounded if and only if it has the property that N can be partitioned into $N_0,N_1,\ldots,N_m$ for some $m\geqslant 0$ so that, for each N_i and each production $A\to\alpha$ with $A\in N_i$, either $\alpha\in(\Sigma\cup N_0\cup\cdots\cup N_{i-1})^*$ or $\alpha\in\Sigma^*N_i\Sigma^*$ [9]. Suppose that G is in double Greibach form. Then, each production $A\to\alpha$ with $A\in N_0$ is in double Greibach operator form. Now, all productions $A\to\alpha$ with $A\in N_1$ can be transformed into the double Greibach operator form by using production substitutions. This task can be repeated for $N_2,N_3,\ldots,N_m$ in sequence to obtain a grammar G' in double Greibach operator form such that $L(G')=L(G)$. As production substitutions clearly preserve nonterminal-boundedness (check that G' satisfies the condition given in the above characterization), it follows that NB-DG $=$ NB-DGO.

For k-Lin $=$ k-Lin-O, $k\geqslant 1$, let $G=(N,\Sigma,P,S)$ be an arbitrary k-linear grammar. Then every linear production in G is in the operator form. Thus, along the above observation for k-Lin-DG $=$ k-Lin-DGO, it is sufficient to transform the production of the form $S\to\alpha$ into the operator form. First, $S\to\alpha$ can be placed in the operator form by using nonterminals of the form (A,a), $A \in N$ and $a \in \Sigma$, as in the proof of Theorem 2.1. Now, to take care of the nonterminals of the form (A,a), we can simply add the following productions, where $A \in N - \{S\}$:

$$(A,a) \to \alpha \qquad \text{if } A \to \alpha a \text{ is in } P,$$
$$(A,a) \to \alpha(B,a) \quad \text{if } A \to \alpha B \text{ is in } P.$$

This may create chain productions (of the form $A \to B$, where A, B are nonterminals), but they can be removed easily by using production substitutions. It is straightforward to see that the resulting grammar G' is a k-linear grammar in operator form and $L(G') = L(G)$. The relation NB = NB-O can be proved by following the proof for NB-DG = NB-DGO, but by using the technique for showing k-Lin = k-Lin-O given above instead of production substitutions.

Finally, the proper inclusions in (1) and (2) follow from the relations in Theorems 3.2 and 3.5. □

Theorem 3.7. DB = DB-DGO.

Proof. It is sufficient to prove that the transformation for double Greibach operator normal form as used in the proof of Theorem 2.1 (together with the Engelfriet's transformation) preserves derivation-boundedness. For this, let $G = (N, \Sigma, P, S)$ be a derivation-bounded context-free grammar. It is easy to observe that derivation-boundedness is preserved when the Chomsky normal-form transformation is applied to a context-free grammar. So, we can assume that G is a derivation-bounded grammar in Chomsky normal form. One can also observe that the production substitution and λ-production removal preserve derivation-boundedness. (The latter case is trivial. For the former, let $\bar{G}$ be a derivation-bounded grammar with a constant k, the maximum number of nonterminals in a sentential form in any optimal derivation, and let l be the maximum number of nonterminals in the right-hand sides of the productions of $\bar{G}$. Suppose that $\bar{G}'$ is obtained from $\bar{G}$ by applying any production substitution. It is not difficult to see that each word $w \in L(\bar{G})$ has a derivation in $\bar{G}'$ such that each sentential form contains at most kl nonterminals.) Therefore, it is sufficient to prove now that the grammar G' containing productions (1)–(8), produced by the Engelfriet's transformation as an intermediate grammar (see Section 2), is derivation-bounded. This is because the Engelfriet's transformation starts with a grammar in Chomsky normal form, the rest of the Engelfriet's transformation applied to (1)–(8) use production substitutions and λ-production removals only, and the further transformation into the double Greibach operator form (given in the proof of Theorem 2.1) simply repeats a special case of the Engelfriet's transformation.

Let ξ be a derivation in G and let $A \Rightarrow_G^* \alpha$, where $A \in N$ and $\alpha \in (N \cup \Sigma)^*$. For each word $\beta \in (N \cup \Sigma)^*$, let $\#_N(\beta)$ denote the number of nonterminals in β. In the sequel, we shall assume without loss of generality that every nonterminal of G is useful, in that it is used in a derivation for some terminal word. Define the following:

$$\phi_G(\xi) = \max\{\#_N(\beta) \mid \beta \text{ is a sentential form in } \xi\},$$

$$\phi_G(A,\alpha) = \min\{\phi_G(\mu) \mid \mu : A \Rightarrow_G^* \alpha\},$$

$$\phi_G(A) = \max\{\phi_G(A,w) \mid A \Rightarrow_G^* w\,(\in \Sigma^*)\}.$$

Observe that $\phi_G(S)$, called the index of G, determines the integer k in the definition of a derivation-bounded grammar G. For the grammar G' produced by the Engelfriet's transformation, define $\phi_{G'}(\xi)$, $\phi_{G'}(A,\alpha)$ and $\phi_{G'}(A)$ analogously.

Let A be an arbitrary nonterminal from N and let w be an arbitrary terminal word such that $A \Rightarrow_G^* w$. Let $\xi : A \Rightarrow_G^* w$ be a derivation such that $\phi_G(\xi) = \phi_G(A,w)$. Let D_1 be any nonterminal from N such that ξ has the form $A \Rightarrow_G^* D_1\alpha \Rightarrow_G^* w_1w_1'$ $(=w)$, where $D_1 \Rightarrow_G^* w_1$ and $\alpha \in (N \cup \Sigma)^*$. Let D_2 be any nonterminal from N such that ξ has the form $A \Rightarrow_G^* \beta D_2 \Rightarrow_G^* w_2'w_2$ $(=w)$, where $\beta \in (N \cup \Sigma)^*$ and $D_2 \Rightarrow_G^* w_2$. (It is possible, as a special case, that $D_1\alpha$ or βD_2 is identical to A.) Recall now relations (i)–(iii) given in Section 2 that illustrate the Engelfriet's transformation, which imply the following relations: (i′) $A \Rightarrow_{G'}^* w$, (ii′) $[D_1,A] \Rightarrow_{G'}^* w_1'$, and (iii′) $(A,D_2) \Rightarrow_{G'}^* w_2'$. It is sufficient to prove that

(1) $\phi_{G'}(A,w) \leqslant (\phi_G(A,w))^2$,
(2) $\phi_{G'}([D_1,A],w_1') \leqslant (\phi_G(A,w))^2$,
(3) $\phi_{G'}((A,D_2),w_2') \leqslant (\phi_G(A,w))^2$,

since the first relation immediately implies that $\phi_{G'}(S) \leqslant (\phi_G(S))^2$, and so, G' is derivation-bounded. We shall prove these three relations by an induction on the length of w.

The induction basis is the case where $A \Rightarrow_G w$ $(\in \Sigma)$, and $\phi_G(A,w) = 1$ in this case. First, relation (1) holds since $A \Rightarrow_{G'} w$, and so, $\phi_{G'}(A,w) = 1$. For relations (2) and (3), it must be that $D_1\alpha = \beta D_2 = A$, and so, $w_1' = w_2' = \lambda$. Then, $\phi_{G'}([D_1,A],w_1') = \phi_{G'}((A,D_2),w_2') = 1$, and so, relations (2) and (3) hold. Therefore, the induction basis holds true.

Assume now that relations (1)–(3) hold for all $w \in \Sigma^*$ with $\#(w) \leqslant n_0$, for some $n_0 \geqslant 1$. We shall prove that these relations hold when $\#(w) = n_0 + 1$.

Consider first relation (1). As $\#(w) \geqslant 2$ and G is in Chomsky normal form, ξ has the form $A \Rightarrow_G B_1B_2 \Rightarrow_G^* ax_1x_2b$ $(=w)$, where $a,b \in \Sigma$, $B_1 \Rightarrow_G^* ax_1$ and $B_2 \Rightarrow_G^* x_2b$. Furthermore, the first and last symbols of w must be generated by using productions of the form $Y \to a$ and $Z \to b$ in ξ. This means that there is a derivation $A \Rightarrow_{G'} a[Y,B_1](B_2,Z)b \Rightarrow_{G'}^* ax_1x_2b$ $(=w)$, where $[Y,B_1] \Rightarrow_{G'}^* x_1$ and $(B_2,Z) \Rightarrow_{G'}^* x_2$. Note that $\#(ax_1)$, $\#(x_2b) \leqslant n_0$. Therefore, by the induction hypothesis, the following relations hold:

$$\phi_{G'}([Y,B_1],x_1) \leqslant (\phi_G(B_1,ax_1))^2,$$

$$\phi_{G'}((B_2,Z),x_2) \leqslant (\phi_G(B_2,x_2b))^2.$$

Now, because of the structure of ξ (consider the corresponding derivation tree), one of the following relations must hold:

$$\phi_G(A,w) \geqslant \begin{cases} \phi_G(B_2,x_2b) & \text{if } \phi_G(B_1,ax_1) < \phi_G(B_2,x_2b), \\ \phi_G(B_1,ax_1)+1 & \text{if } \phi_G(B_1,ax_1) = \phi_G(B_2,x_2b), \\ \phi_G(B_1,ax_1) & \text{if } \phi_G(B_1,ax_1) > \phi_G(B_2,x_2b). \end{cases}$$

Suppose that $\phi_G(B_1,ax_1) < \phi_G(B_2,x_2b)$. Let μ be the derivation $A \Rightarrow_{G'} a[Y,B_1](B_2,Z)b \Rightarrow^*_{G'} ax_1(B_2,Z)b \Rightarrow^*_{G'} ax_1x_2b$ $(=w)$. Then, the following relations prove the induction step:

$$\begin{aligned} \phi_{G'}(\mu) &= \max\{\phi_{G'}([Y,B_1],x_1)+1, \phi_{G'}((B_2,Z),x_2)\} \\ &\leqslant \max\{(\phi_G(B_1,ax_1))^2+1, (\phi_G(B_2,x_2b))^2\} \\ &= (\phi_G(B_2,x_2b))^2 \\ &\leqslant (\phi_G(A,w))^2. \end{aligned}$$

It is not difficult to see that, if $\phi_G(B_1,ax_1) = \phi_G(B_2,x_2b)$, then the same μ as above yields the relation $\phi_{G'}(\mu) \leqslant (\phi_G(A,w))^2$. Now, if $\phi_G(B_1,ax_1) > \phi_G(B_2,x_2b)$, then consideration of the derivation in G' that executes $(B_2,Z) \Rightarrow^*_{G'} x_2$ first and then $[Y,B_1] \Rightarrow^*_{G'} x_1$ yields the target relation. This completes the induction step for relation (1).

Consider now relation (2). First, if $D_1\alpha = A$ (and so, $\#(w_1) \geqslant 2$ and $w_1' = \lambda$) then $\phi_{G'}([D_1,A],w_1') = 1$ and relation (2) clearly holds. So, assume that $D_1\alpha \neq A$, and so, $w_1 \neq \lambda$ and $w_1' \neq \lambda$ since G is in Chomsky normal form. Then, ξ must be either of the form $A \Rightarrow_G D_1R \Rightarrow^*_G D_1w_1'$ or of the form $A \Rightarrow_G BR \Rightarrow^*_G D_1xyz$ $(=w)$, where $B \Rightarrow^*_G C\gamma \Rightarrow_G D_1Q\gamma$, $Q \Rightarrow^*_G x$, $\gamma \Rightarrow^*_G y$, and $R \Rightarrow^*_G z$.

For the former case, there is a derivation $\mu : [D_1,A] \Rightarrow_{G'} R \Rightarrow^*_{G'} w_1'$ and the following relations prove the induction step:

$$\phi_{G'}(\mu) = \phi_{G'}(R,w_1') \leqslant (\phi_G(R,w_1'))^2 \leqslant (\phi_G(A,w))^2,$$

where the second relation follows from the induction hypothesis (since $\#(w_1') \leqslant n_0$) and the third relation is obvious from the structure of ξ.

For the latter case, note first that there is a derivation $[D_1,A] \Rightarrow_{G'} Q[C,B]R \Rightarrow^*_{G'} xyz$, where $Q \Rightarrow^*_{G'} x$, $[C,B] \Rightarrow^*_{G'} y$, and $R \Rightarrow^*_{G'} z$. Clearly, x and z are nonempty words. Therefore, $\#(x), \#(w_1xy), \#(z) \leqslant n_0$, and so, the induction hypothesis implies the following relations:

$$\phi_{G'}(Q,x) \leqslant (\phi_G(Q,x))^2,$$

$$\phi_{G'}([C,B],y) \leqslant (\phi_G(B,w_1xy))^2,$$

$$\phi_{G'}(R,z) \leqslant (\phi_G(R,z))^2.$$

Furthermore, the structure of ξ implies that $\phi_G(Q,x) \leqslant \phi_G(B,w_1xy)$ and

$$\phi_G(A,w) \geqslant \begin{cases} \phi_G(R,z) & \text{if } \phi_G(B,w_1xy) < \phi_G(R,z), \\ \phi_G(B,w_1xy)+1 & \text{if } \phi_G(B,w_1xy) = \phi_G(R,z), \\ \phi_G(B,w_1xy) & \text{if } \phi_G(B,w_1xy) > \phi_G(R,z). \end{cases}$$

Let us first consider the case where $\phi_G(B,w_1xy) \leqslant \phi_G(R,z)$. There is a derivation μ : $[D_1,A] \Rightarrow_{G'} Q[C,B]R \Rightarrow^*_{G'} x[C,B]R \Rightarrow^*_{G'} xyR \Rightarrow^*_{G'} xyz$. Then,

$$\begin{aligned} \phi_{G'}(\mu) &= \max\{\phi_{G'}(Q,x)+2, \phi_{G'}([C,B],y)+1, \phi_{G'}(R,z)\} \\ &\leqslant \max\{(\phi_G(Q,x))^2+2, (\phi_G(B,w_1xy))^2+1, (\phi_G(R,z))^2\} \\ &\leqslant \max\{(\phi_G(B,w_1xy))^2+2, (\phi_G(R,z))^2\}, \end{aligned}$$

which can be easily seen to be no larger than $(\phi_G(A,w))^2$ for both the cases $\phi_G(B,w_1xy) < \phi_G(R,z)$ and $\phi_G(B,w_1xy) = \phi_G(R,z)$.

Consider now the case where $\phi_G(B,w_1xy) > \phi_G(R,z)$. If $\phi_G(Q,x) < \phi_G(B,w_1xy)$, then it is easy to see that the derivation μ : $[D_1,A] \Rightarrow_{G'} Q[C,B]R \Rightarrow^*_{G'} x[C,B]R \Rightarrow^*_{G'} x[C,B]z \Rightarrow^*_{G'} xyz$ yields the relation $\phi_{G'}(\mu) \leqslant (\phi_G(A,w))^2$. So, assume that $\phi_G(Q,x) = \phi_G(B,w_1xy)$. Suppose that $B \Rightarrow^*_G C\gamma$ in ξ has the form

$$\begin{aligned} B &\Rightarrow_G B_1R_1 \\ &\Rightarrow^*_G B_1\gamma_1^{(1)} \\ &\Rightarrow_G B_2R_2\gamma_1^{(1)} \\ &\Rightarrow^*_G B_2\gamma_2^{(1)}\gamma_1^{(2)} \\ &\Rightarrow_G B_3R_3\gamma_2^{(1)}\gamma_1^{(2)} \\ &\Rightarrow^*_G B_3\gamma_3^{(1)}\gamma_2^{(2)}\gamma_1^{(3)} \\ &\Rightarrow^*_G B_mR_m\gamma_{m-1}^{(1)}\cdots\gamma_1^{(m-1)} \\ &\Rightarrow^*_G B_m\gamma_m^{(1)}\cdots\gamma_1^{(m)} \\ &\Rightarrow^*_G C_mR_{m+1}\gamma_m^{(1)}\cdots\gamma_1^{(m)} \\ &\Rightarrow^*_G C_m\gamma_{m+1}^{(1)}\gamma_m^{(2)}\cdots\gamma_1^{(m+1)} \quad (= C\gamma). \end{aligned}$$

Let $\gamma_i^{(m-i+2)} \Rightarrow^*_G y_i$ $(\in \Sigma^*)$ in ξ for all $i \in \{1,2,\ldots,m+1\}$. Thus, $y = y_{m+1}y_m\cdots y_1$. There is a derivation $\bar{\mu}$ in G' with the following form:

$$\begin{aligned} [C,B] &\Rightarrow_{G'} R_{m+1}[B_m,B_1]R_1 \\ &\Rightarrow^*_{G'} R_{m+1}[B_m,B_1]y_1 \\ &\Rightarrow^*_{G'} y_{m+1}[B_m,B_1]y_1 \\ &\Rightarrow_{G'} y_{m+1}R_m[B_{m-1},B_2]R_2y_1 \end{aligned}$$

$$
\begin{aligned}
&\Rightarrow^*_{G'} y_{m+1}R_m[B_{m-1},B_2]y_2y_1\\
&\Rightarrow^*_{G'} y_{m+1}y_m[B_{m-1},B_2]y_2y_1\\
&\Rightarrow^*_{G'} y_{m+1}y_m\cdots y_{i+1}[B_i,B_j]y_jy_{j-1}\cdots y_1\\
&\Rightarrow_{G'} y_{m+1}y_m\cdots y_{i+1}\delta y_jy_{j-1}\cdots y_1\\
&\Rightarrow^*_{G'} y_{m+1}y_m\cdots y_1\ (=y),
\end{aligned}
$$

where $i+j=m+1$ and either $i=j$ and $\delta=\lambda$ or $i=j+1$ and $\delta=R_i$. Note that, as $\phi_G(Q,x)=\phi_G(B,w_1xy)$, it must be that $\phi_G(R_i,y_i) < \phi_G(Q,x)$ for all $i\in\{1,2,\dots,m+1\}$. Then,

$$
\begin{aligned}
\phi_{G'}(\bar{\mu}) &= \max\{\phi_{G'}(R_i,y_i)+2,\phi_{G'}(R_{m-i+2},y_{m-i+2})+1 \mid 1\leqslant i\leqslant \lceil m/2\rceil\}\\
&\leqslant \max\{(\phi_G(R_i,y_i))^2+2 \mid 1\leqslant i\leqslant m+1\}\\
&< (\phi_G(Q,x))^2\\
&= (\phi_G(B,w_1xy))^2,
\end{aligned}
$$

where the second relation holds by the induction hypothesis. Now, let μ be the derivation $[D_1,A]\Rightarrow_{G'} Q[C,B]R\Rightarrow^*_{G'} Q[C,B]z\Rightarrow^*_{G'} Qyz\Rightarrow^*_{G'} xyz$, where $[C,B]\Rightarrow^*_{G'} y$ in μ is obtained by using $\bar{\mu}$. Then,

$$
\begin{aligned}
\phi_{G'}(\mu) &= \max\{\phi_{G'}(R,z)+2,\phi_{G'}(\bar{\mu})+1,\phi_{G'}(Q,x)\}\\
&\leqslant \max\{(\phi_G(R,z))^2+2,(\phi_G(B,w_1xy))^2,(\phi_G(Q,x))^2\}\\
&= (\phi_G(B,w_1xy))^2\\
&\leqslant (\phi_G(A,w))^2.
\end{aligned}
$$

This completes the induction step for relation (2). The induction step for relation (3) can be proved analogously. Now, this completes the induction for relations (1)–(3), and it follows that the theorem holds. □

Theorem 3.8. *The inclusion relations shown in Fig.* 1 *are true, where arrows denote proper inclusions and two classes not related by a chain of arrows are incomparable.*

Proof. Theorems 3.2, 3.4 and 3.5 imply the incomparability relations and the relations Lin-DGO $\subset$ 2-Lin-DGO $\subset$ 3-Lin-DGO $\subset\cdots\subset$ NB-DGO in Fig. 1. Altogether, the present theorem follows from Theorems 2.1 and 3.2–3.7. □

As the final analysis in this section, we shall characterize the classes k-Lin, $k\geqslant 1$, and NB by their corresponding DGO subclasses via homomorphisms. Note that the DGO classes are closed under nonerasing homomorphisms since each terminal symbol in the right-hand side of a production can be substituted by its homomorphic image. Thus, Theorem 3.8 and the following theorem imply that, for example, Lin and Lin-DGO are different because of the nonclosure of Lin-DGO under homomorphisms with erasing.

Theorem 3.9. *For every language $L \in k$-Lin, $k \geqslant 1$, there exist a language $L' \in$ k-*Lin-DGO *and a homomorphism h such that $h(L') = L$. The same statement holds for $L \in$* NB *and $L' \in$* NB-DGO.

Proof. Let G be a linear grammar in $(A \to aB, A \to Ba, A \to a)$-normal form and let ¢ be a terminal symbol not used by G. Let G' be obtained from G by modifying $A \to aB$ to $A \to aB¢$ and $A \to Ba$ to $A \to ¢Ba$. Then, G' is a Lin-DGO grammar. Let h be a homomorphism that erases all ¢'s and preserves other symbols. Clearly, $h(L(G')) = L(G)$. The cases for k-Lin, $k \geqslant 2$, and NB are similar. □

4. Decision properties

We shall consider several fundamental language-theoretic decision properties, which have been well studied for existing context-free subclasses. Our goal in this section is to analyze them for new classes contained in the hierarchy constructed in the previous section. The results proved in this section can be summarized as follows (the known results mentioned below can be found in [9–11, 19]):

(1) The problem "$= \Sigma^*$" is known to be undecidable for Lin and decidable for RL. This problem is shown to be undecidable for Lin-DGO and decidable for so-called balanced Lin-DGO (that properly includes RL).

(2) The problems "$L_1 = L_2$" and "$L_1 \subseteq L_2$" are known to be undecidable for a fixed $L_1 \in$ RL and $L_2 \in$ Lin and decidable for $L_1, L_2 \in$ RL. (In fact, "$L_1 \subseteq L_2$" is decidable for $L_1 \in$ CF and $L_2 \in$ RL.) These problems are shown to be undecidable for a fixed $L_1 \in$ RL and $L_2 \in$ Lin-DGO and decidable for $L_1, L_2 \in$ balanced-Lin-DGO.

(3) The problem "$L_1 \cap L_2 = \emptyset$" is undecidable for $L_1 \in$ Lin and $L_2 \in$ 2-Lin (as implicit in the proof of undecidability of this problem for $L_1, L_2 \in$ CF, given in [9, 19]) and decidable for $L_1 \in$ RL and $L_2 \in$ CF. This problem is shown to be undecidable for $L_1, L_2 \in$ Lin-DGO such that one of L_1, L_2 is balanced and decidable for $L_1, L_2 \in$ balanced-Lin-DGO.

Theorem 4.1. *Let Σ be an alphabet. It is undecidable whether or not $L = \Sigma^*$ for a* Lin-DGO *language L.*

Proof. A well-known method for showing undecidability of "$L = \Sigma^*$" for $L \in$ Lin is to describe the set of all "invalid computations" of a Turing machine M by a linear grammar G, so that $L(M) = \emptyset$ if and only if $L(G) = \Sigma^*$ [10]. (The emptiness problem for Turing machines is undecidable.) We shall use a similar method, but by a reduction from the empty-word acceptance problem for Turing machines, which is also undecidable.

Let $M = (Q, \Omega, \Gamma, \delta, q_0, \sqcup, F)$ be a Turing machine with a semi-infinite tape which is infinite to the right, where Q is the set of states, Ω is the input alphabet, Γ is the total tape alphabet, $\delta : Q \times \Gamma \to 2^{Q \times \Gamma \times \{l, r\}}$ is the transition function (l and r denote the left and right moves), $q_0 \in Q$ is the initial state, $\sqcup \in \Gamma$ is the blank symbol, and $F \subseteq Q$ is the set of accepting states. We shall assume without loss of generality that M does not print a blank symbol.

A *configuration* of M is a word of the form $xqy\sqcup$, where $xy \in (\Gamma - \{\sqcup\})^*$ and $q \in Q$. (We assume $Q \cap \Gamma = \emptyset$.) Intuitively, $xqy\sqcup$ describes the situation of M such that the nonblank tape content is xy and M is scanning the leftmost symbol of $y\sqcup$ in the state q. The *initial configuration* is $q_0\sqcup$ and an *accepting configuration* is any configuration $xqy\sqcup$ with $q \in F$. For configurations I and J, denote by $I \vdash J$ if M in I can go to J in one step. Let ¢ be a symbol not in $\Gamma \cup Q$. A *valid computation* of M is a word of the form

$$¢w_1¢w_2¢ \cdots ¢w_n¢¢w_n^R¢ \cdots ¢w_2^R¢w_1^R¢,$$

where $n \geqslant 1$ and w_i^R denotes the reversal of w_i, such that (i) each w_i is a configuration of M, (ii) w_1 is the initial configuration, (iii) w_n is an accepting configuration, and (iv) $w_i \vdash w_{i+1}$ for all $i = 1, 2, \ldots, n-1$. Clearly, $\lambda \in L(M)$ if and only if there exists a valid computation of M.

Now, let $\Sigma = \Gamma \cup Q \cup \{¢\}$. Then, a word $w \in \Sigma^*$ is an invalid computation of M if at least one of the following conditions holds:

(1) w is not of the form $¢x_1¢ \cdots ¢x_n¢¢y_m¢ \cdots ¢y_1¢$ $(n, m \geqslant 1)$, where all x_i and y_i are configurations of M (otherwise, assume such a form for w in the conditions listed below),

(2) $x_1 \neq q_0\sqcup$,

(3) $x_n \notin (\Gamma - \{\sqcup\})^* F (\Gamma - \{\sqcup\})^* \{\sqcup\}$,

(4) w is not an even-length palindrome,

(5) $y_1 = \sqcup q_0$ and $\#(y_{i+1}) - \#(x_i) \notin \{0, 1\}$ for some i, or

(6) $y_1 = \sqcup q_0$ and $x_i \vdash y_{i+1}^R$ is false for some i, where $\#(y_{j+1}) - \#(x_j) \in \{0, 1\}$ for all $j < i$.

The set of all words satisfying (1), (2) or (3) is a language in RL. It can be easily seen that the set of all words satisfying (4) is a language in Lin-DGO. For (5), consider the Lin-DGO grammar $G_1 = (\{S_1, A_1, B_1\}, \Sigma, P_1, S_1)$ such that P_1 consists of the following productions, where $a, b \in \Sigma - \{¢\}$ and $c, d \in \Sigma$:

$$S_1 \to ¢\, A_1\, ¢\, \sqcup\, q_0\, ¢,$$

$$A_1 \to aA_1b \mid ¢A_1¢ \mid ¢A_1¢b \mid ¢B_1ab,$$

$$B_1 \to cB_1d | c | cd.$$

Observe that A_1 finds the smallest i such that $\#(y_{i+1}) - \#(x_i) \notin \{0, 1\}$. It is easy to see that G_1 generates invalid computations only, including all words satisfying (5). Now, for (6), consider the Lin-DGO grammar $G_2 = (\{S_2, A_2, B_2\}, \Sigma, P_2, S_2)$ such that P_2 consists of the following productions, where $a, b \in \Sigma - \{¢\}$, $c, d \in \Gamma$, and $e, f \in \Sigma$:

$$S_2 \to ¢A_2¢ \sqcup q_0¢,$$

$$A_2 \to aA_2b \mid ¢A_2¢ \mid ¢A_2¢b,$$

$A_2 \to pcB_2qd$ if $(q,d,r) \notin \delta(p,c)$,

$A_2 \to apcB_2daq$ if $(q,d,l) \notin \delta(p,c)$,

$B_2 \to eB_2f \mid e \mid ef$.

Observe that A_2 finds the smallest i such that $x_i \vdash y_{i+1}^R$ is false and that G_2 generates invalid computations only, including all words satisfying (6).

As RL $\subseteq$ Lin-DGO and Lin-DGO is clearly closed under union, there exists a Lin-DGO grammar G generating all invalid computations of M. Then, $\lambda \notin L(M)$ if and only if $L(G) = \Sigma^*$. So, the theorem holds. $\square$

A language is *bounded* if it is a subset of $w_1^* w_2^* \cdots w_n^*$ for some fixed words $w_1, w_2, \ldots, w_n$; otherwise, it is *unbounded.* Hunt and Rosenkrantz [11] showed that the problems "$L_1 = L_2$" and "$L_1 \subseteq L_2$" are undecidable for $L_1 \in$ RL and $L_2 \in$ Lin, where L_1 is a fixed unbounded language, by proving a more general result stated in the following lemma.

Lemma 4.2. *Let $\mathscr{F}$ be any effective family of languages that is effectively closed under* 1-1 *homomorphisms, union, and concatenation with singletons. Let $\mathscr{F}$ be also closed* (*not necessarily effectively*) *under intersection with regular languages. If* "$= \Sigma^*$" *is undecidable for $\mathscr{F}$, then so are* "$= L_0$" *and* "$\supseteq L_0$" *for all $L_0 \in \mathscr{F}$ containing an unbounded regular subset.*

Theorem 4.3. *Let L_1 be an arbitrary fixed unbounded* RL *language. It is undecidable whether or not* (1) $L_1 = L_2$, *and* (2) $L_1 \subseteq L_2$, *for a* Lin-DGO *language* L_2.

Proof. Note first that the 1-1 homomorphisms in Lemma 4.2 can be replaced by nonerasing 1-1 homomorphisms; this is implicit in the proof of the result in Lemma 4.2 in [11] (see Theorem 2.2 and Corollary 2.3). It is easy to see that Lin-DGO is effectively closed under nonerasing 1-1 homomorphisms, union, and concatenation with singletons. As "$= \Sigma^*$" is undecidable for Lin-DGO (Theorem 4.1), it is sufficient to prove that Lin-DGO is closed under intersection with regular languages.

Suppose that $G_1 = (N_1, \Sigma_1, P_1, S_1)$ is a Lin-DGO grammar and $G_2 = (N_2, \Sigma_2, P_2, S_2)$ is an RL grammar. Construct a Lin-DGO grammar $G = ((N_2 \times N_1 \times N_2) \cup \{S_1\}, \Sigma_1, P, S_1)$ such that P consists of the following productions, where $A, B \in N_1$, $Q, R, \bar{Q}, \bar{R} \in N_2$ and $x, y \in \Sigma_1^*$:

$S_1 \to x$ if $S_1 \to x$ is in P_1 and $S_2 \Rightarrow_{G_2}^* x$,

$S_1 \to x[Q,A,R]y$ if $S_1 \to xAy$ is in $P_1, S_2 \Rightarrow_{G_2}^* xQ$, and $R \Rightarrow_{G_2}^* y$,

$[Q,A,R] \to x[\bar{Q},B,\bar{R}]y$ if $A \to xBy$ is in $P_1, Q \Rightarrow_{G_2}^* x\bar{Q}$, and $\bar{R} \Rightarrow_{G_2}^* yR$,

$[Q,A,R] \to x$ if $A \to x$ is in P_1 and $Q \Rightarrow_{G_2}^* xR$.

G has been constructed by using a two-way cross product. It should not be difficult to see that $L(G) = L(G_1) \cap L(G_2)$. Thus, Lin-DGO is (effectively) closed under intersection with regular languages. Now, the theorem follows from Lemma 4.2. □

Definition 4.4. A Lin-DGO grammar $G = (N, \Sigma, P, S)$ is a *balanced grammar* if each production of the form $A \to xBy$, $A, B \in N$ and $x, y \in \Sigma^*$, has the property that $\#(x) = \#(y)$. The class of languages generated by balanced Lin-DGO grammars is denoted by balanced-Lin-DGO.

Theorem 4.5. RL $\subset$ balanced-Lin-DGO $\subset$ Lin-DGO.

Proof. The proof for RL $\subset$ Lin-DGO (Theorem 3.3) works for RL $\subset$ balanced-Lin-DGO, as it is, and the relation balanced-Lin-DGO $\subset$ Lin-DGO follows from the inclusion undecidability in Theorem 4.3 and the decidability result to be proved in the next theorem. □

Theorem 4.6. *It is decidable whether or not* (1) $L_1 \subseteq L_2$, *and* (2) $L_1 \cap L_2 = \emptyset$, *for balanced* Lin-DGO *languages* L_1 *and* L_2.

Proof. Every balanced Lin-DGO grammar can be easily transformed into one in which each production is of the form $A \to aBb$, $A \to ab$ or $A \to a$, where A, B are nonterminals and a, b are terminals. Let G_1, G_2 be Lin-DGO grammars in such a normal form. For each $i \in \{0, 1\}$, let (G_i) denote the parenthesis version of G_i, whose productions include $A \to (\alpha)$ if $A \to \alpha$ is a production of G_i. Then, $L(G_1) \subseteq L(G_2)$ if and only if $L((G_1)) \subseteq L((G_2))$. As the inclusion problem for parenthesis grammars is decidable [19], the inclusion part of the theorem holds.

Two balanced Lin-DGO grammars G_1, G_2 in $(A \to aBb, A \to ab, A \to a)$-normal form can be easily cross-producted to yield a Lin-DGO grammar G such that $L(G) = L(G_1) \cap L(G_2)$. Now, the intersection-emptiness part of the theorem follows from the fact that the emptiness problem is decidable for CF. □

Corollary 4.7. *The problems* "$= \Sigma^*$" *and* "$L_1 = L_2$" *are decidable for balanced* Lin-DGO *languages.*

Theorem 4.8. *It is undecidable whether or not* $L_1 \cap L_2 = \emptyset$ *for* Lin-DGO *languages* L_1 *and* L_2 *one of which is balanced.*

Proof. Let $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$ be two lists of nonempty words over $\{0, 1\}$, given as an instance of the Post correspondence problem (PCP). Construct a Lin-DGO grammar $G_1 = (\{S\}, \{0, 1, ¢\}, P, S)$ whose productions are $S \to x_i S y_i^R$, $1 \leqslant i \leqslant n$, and $S \to ¢$. It is easy to construct a balanced Lin-DGO grammar G_2 generating the set $\{u¢v \mid u, v \in \{0, 1\}^+, u = v^R\}$. Then, (x, y) is a positive instance of PCP if and only if $L(G_1) \cap L(G_2) \neq \emptyset$. As PCP is undecidable, the theorem holds. □

References

[1] B. Courcelle, J. Engelfriet, G. Rozenberg, Handle-rewriting hypergraph grammars, J. Comput. System Sci. 46 (1993) 218–270.
[2] J. Engelfriet, An elementary proof of double Greibach normal form, Inform. Process. Lett. 44 (1992) 291–293.
[3] J. Engelfriet, L. Heyker, G. Leih, Context-free graph languages of bounded degree are generated by apex graph grammars, Acta Inform. 31 (1994) 341–378.
[4] J. Engelfriet, G. Leih, Linear graph grammars: power and complexity, Inform. Comput. 81 (1989) 88–121.
[5] J. Engelfriet, G. Leih, G. Rozenberg, Apex graph grammars and attribute grammars, Acta Inform. 25 (1988) 537–571.
[6] J. Engelfriet, G. Leih, E. Welzl, Boundary graph grammars with dynamic edge relabeling, J. Comput. System Sci. 40 (1990) 307–345.
[7] R.W. Floyd, Syntactic analysis and operator precedence, J. ACM 10 (1963) 316–333.
[8] S.A. Greibach, A new normal-form theorem for context-free phrase-structure grammars, J. ACM 12 (1965) 42–52.
[9] M.A. Harrison, Introduction to Formal Language Theory, Addison-Wesley, Reading, MA, 1978.
[10] J.E. Hopcroft, J.D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, Reading, MA, 1979.
[11] H.B. Hunt III, D.J. Rosenkrantz, On equivalence and containment problems for formal languages, J. ACM 24 (1977) 387–396.
[12] C. Kim, A hierarchy of eNCE families of graph languages, Theoret. Comput. Sci. 186 (1997) 157–169.
[13] C. Kim, Efficient recognition algorithms for boundary and linear eNCE graph languages, submitted for publication.
[14] C. Kim, T.E. Jeong, HRNCE grammars – A hypergraph generating system with an eNCE way of rewriting, Theoret. Comput. Sci., in press.
[15] C. Kim, D.H. Lee, Separating k-separated eNCE graph languages, Theoret. Comput. Sci. 120 (1993) 247–259.
[16] C. Kim, D.H. Lee, Node replacement graph languages squeezed with chains, trees, and forests, Inform. Comput. 117 (1995) 63–77.
[17] D.J. Rosenkrantz, Matrix equations and normal forms for context-free grammars, J. ACM 14 (1967) 501–507.
[18] G. Rozenberg, E. Welzl, Boundary NLC graph grammars – Basic definitions, normal forms, and complexity, Inform. and Control 69 (1986) 136–167.
[19] A. Salomaa, Formal Languages, Academic Press, New York, 1973.
[20] F.J. Urbanek, On Greibach normal form construction, Theoret. Comput. Sci. 40 (1985) 315–317.